3. Image Enhancement Using Filtering In the Spatial Domain

- Chapter 3 (Image Enhancement in the Spatial Domain)

Filtering for:

- **Removing noise**
- **Sharpening image**
- **Detecting edges**

1-D Linear Time Invariant (LTI) Systems

- **1-D** signal **x**(n) passes through a **system** T: y(n) = T [x(n)]
- A system is **linear:** $T\left[a_1 x_1(n) + a_2 x_2(n)\right] = a_1 y_1(n) + a_2 y_2(n)$
- A system is **time invariant**: $T\left[x(n-m)\right] = y(n-m)$
- An LTI system can be completely characterized by its **impulse response:** $h(n) = T\left[\delta(n)\right]$

1-D convolution $y(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k) = \sum_{k=-\infty}^{\infty} h(k)x(n-k)$

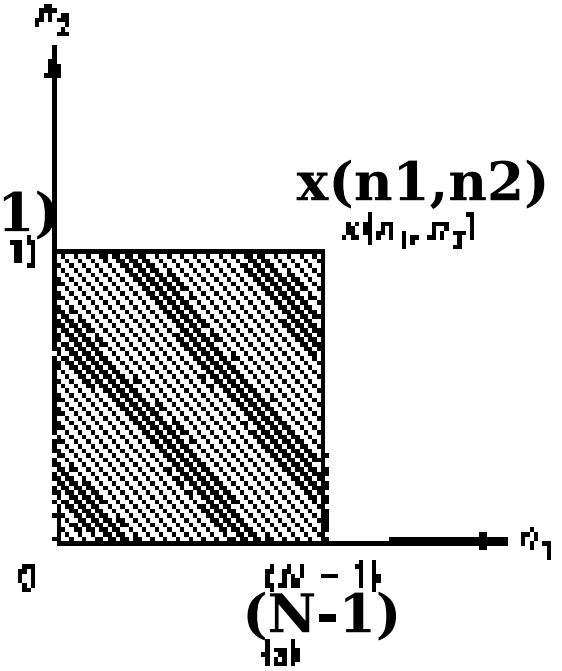
2-D Linear Shift Invariant (LSI) Systems

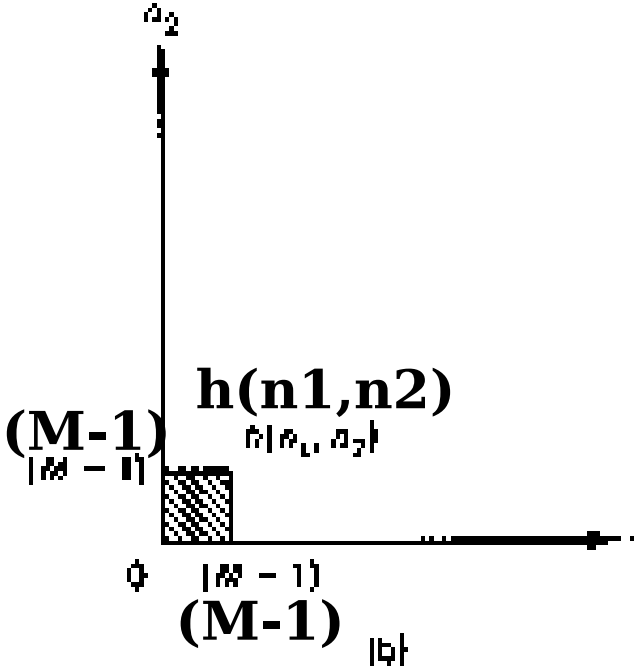
- **2-D** signal $x(n_1, n_2)$ passes through a system T: $y(n_1, n_2) = T[x(n_1, n_2)]$
- A system is **linear**: $T[a_1 x_1(n_1, n_2) + a_2 x_2(n_1, n_2)] = a_1 y_1(n_1, n_2) + a_2 y_2(n_1, n_2)$
- A system is **shift invariant**: $T[x(n_1 - m_1, n_2 - m_2)] = y(n_1 - m_1, n_2 - m_2)$
- The LSI system can be completely characterized by its **unit impulse response** (point spread function): $h(n_1, n_2) = T[\delta(n_1, n_2)]$

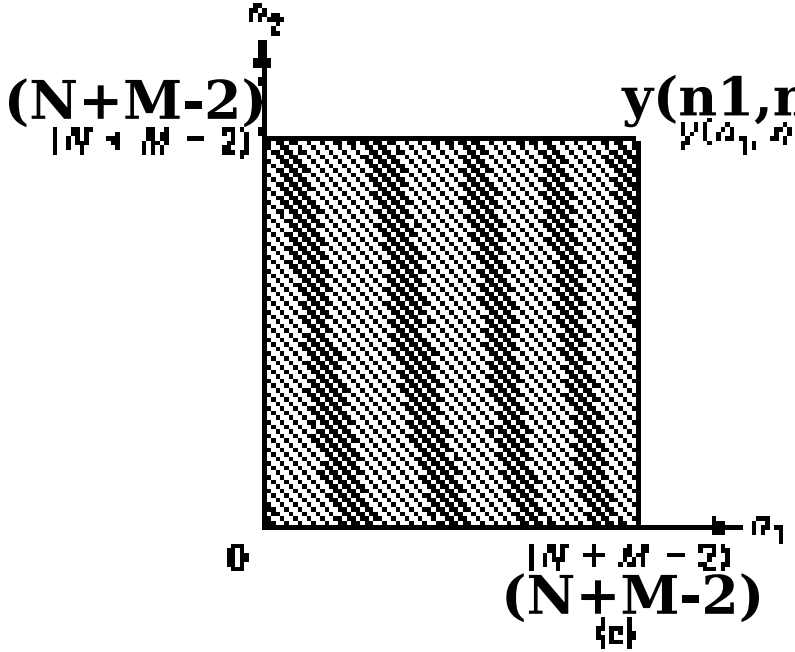
2-D convolution

$$y(n_1, n_2) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x(k_1, k_2) h(n_1 - k_1, n_2 - k_2)$$

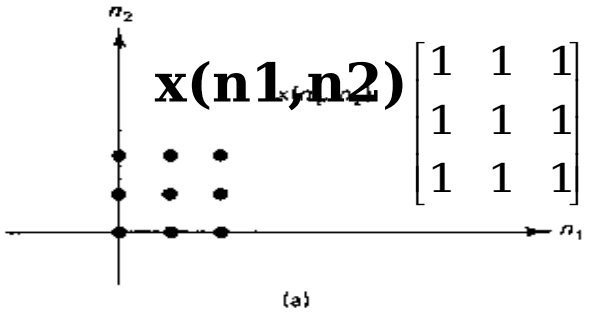
A Typical 2-D Convolution

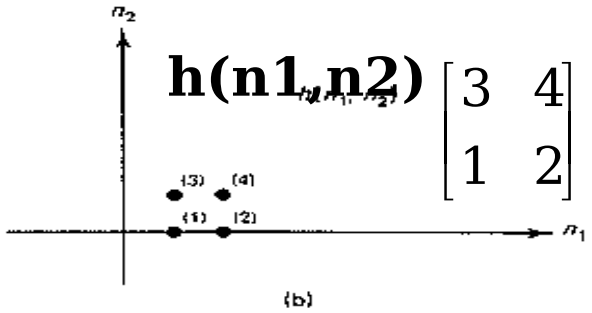


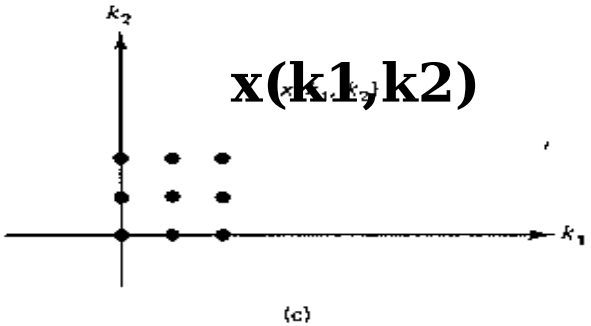


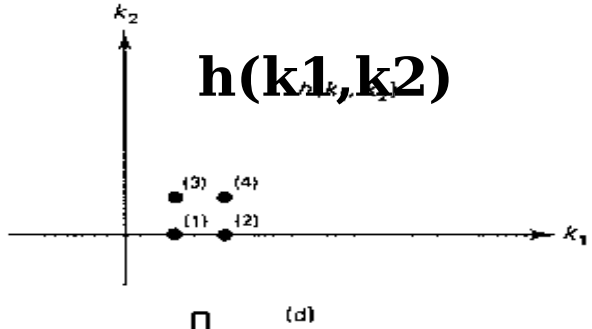


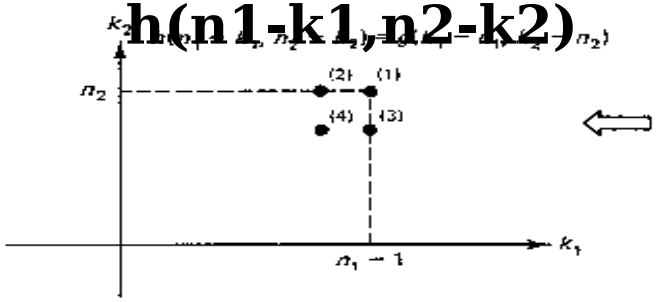
A 2-D Convolution Example

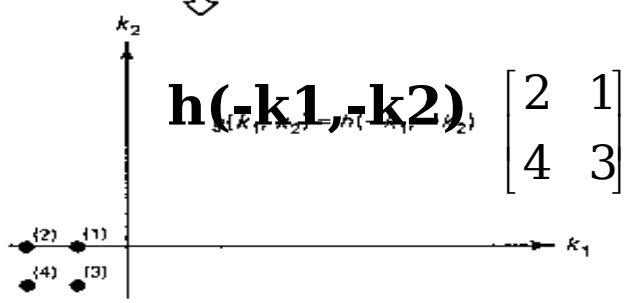


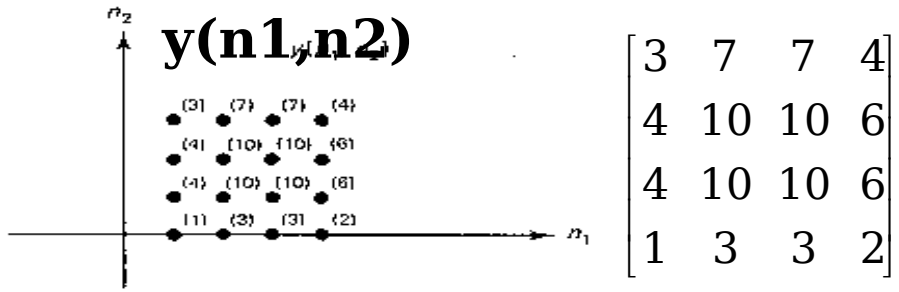












A Separable 2-D System

- Given an M x M system $h(n_1, n_2)$, it is separable if $h(n_1, n_2) = h_1(n_1) h_2(n_2)$ where

$$h_1(n_1) \neq 0, \text{ for } 0 \leq n_1 \leq M-1 \text{ and } h_2(n_2) \neq 0, \text{ for } 0 \leq n_2 \leq M-1$$

- The convolution can be done in two steps:

$$y(n_1, n_2) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} x(k_1, k_2) h_1(n_1 - k_1) h_2(n_2 - k_2)$$

$$= \sum_{k_1=-\infty}^{\infty} h_1(n_1 - k_1) \left[\sum_{k_2=-\infty}^{\infty} x(k_1, k_2) h_2(n_2 - k_2) \right] \quad \textbf{(1)}$$

$$= \sum_{k_1=-\infty}^{\infty} h_1(n_1 - k_1) f(k_1, n_2) \quad \textbf{(2)}$$

Convolution with 2-D Separable Filter

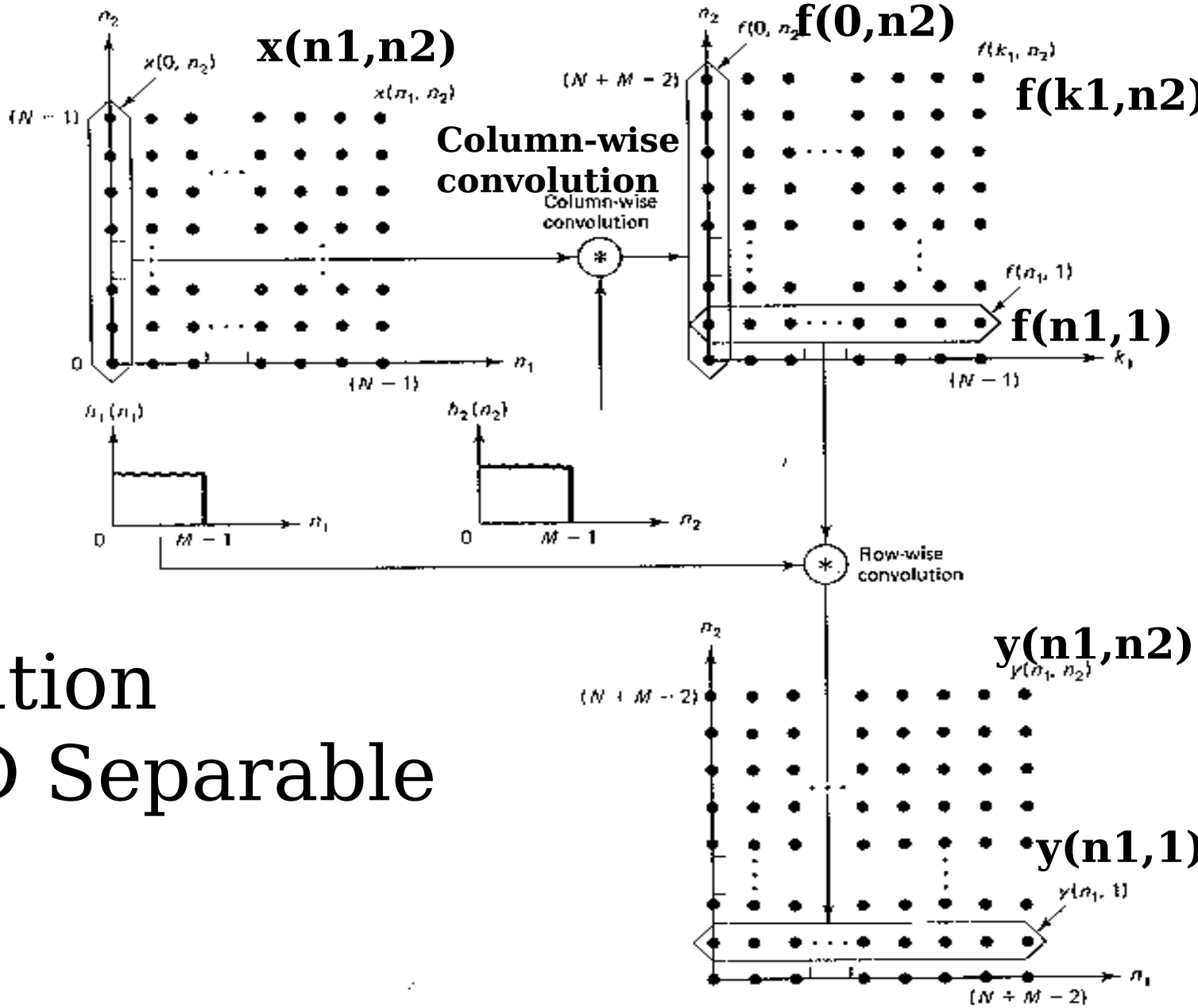
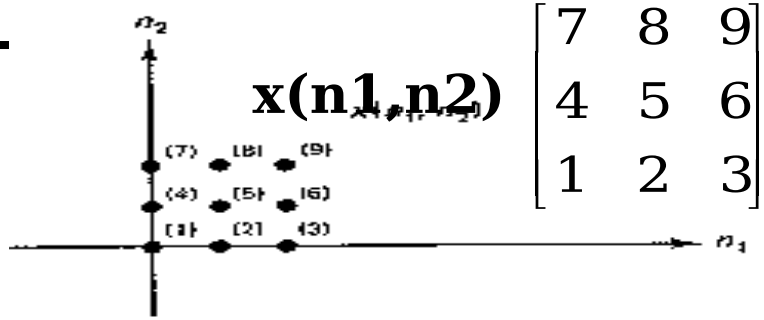


Figure 1.35 Convolution of $x(n_1, n_2)$ with a separable sequence $h(n_1, n_2)$.

Example of 2-D Separable Convolution

x(n1,n2)

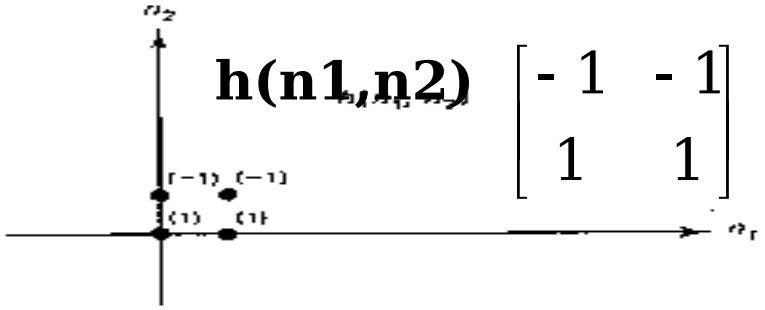
$$\begin{bmatrix} 7 & 8 & 9 \\ 4 & 5 & 6 \\ 1 & 2 & 3 \end{bmatrix}$$



(a)

h(n1,n2)

$$\begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix}$$



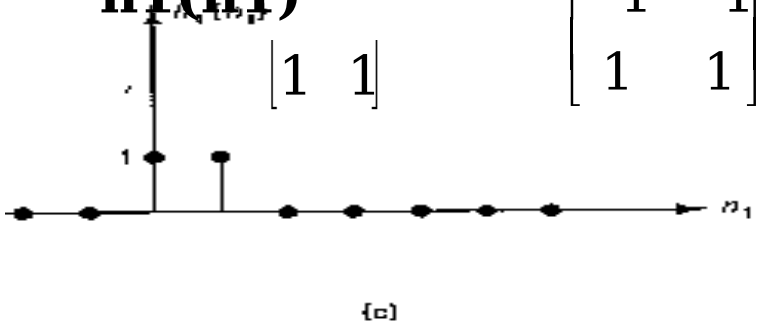
(b)

$$h(n_1,n_2) = h_1(n_1)h_2(n_2)$$

h1(n1)

$$\begin{bmatrix} 1 & 1 \end{bmatrix}$$

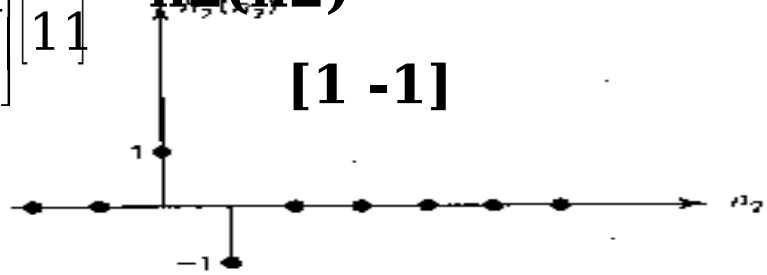
$$\begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \end{bmatrix}$$



(c)

h2(n2)

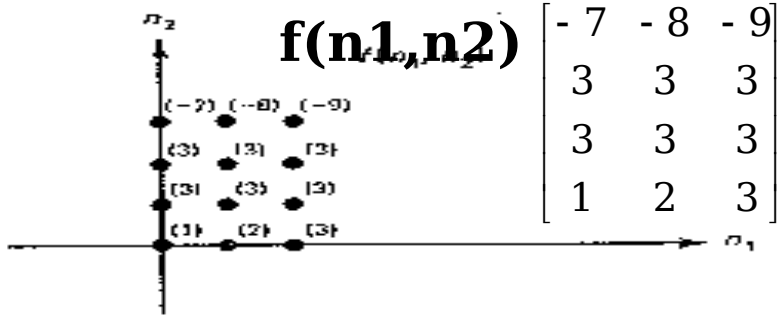
[1 -1]



(d)

f(n1,n2)

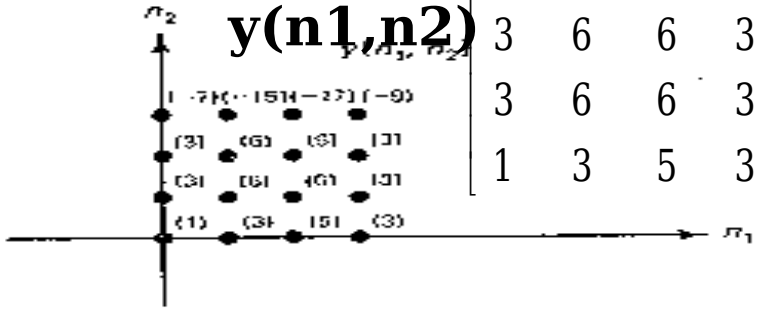
$$\begin{bmatrix} -7 & -8 & -9 \\ 3 & 3 & 3 \\ 3 & 3 & 3 \\ 1 & 2 & 3 \end{bmatrix}$$



(e)

y(n1,n2)

$$\begin{bmatrix} -7 & -15 & -17 & -9 \\ 3 & 6 & 6 & 3 \\ 3 & 6 & 6 & 3 \\ 1 & 3 & 5 & 3 \end{bmatrix}$$



(f)

A Separable 2-D System

(1): for a fixed k_1, we perform 1-D convolution of $x(k_1, n_2)$ with $h_2(n_2)$.

(2): for a fixed n_2, we perform 1-D convolution of $f(k_1, n_2)$ with $h_1(n_1)$.

Total # of multiplications: (for N > > M, reduction facto ~M/2)

$$\sim N^2 M^2 \Rightarrow \sim 2N^2 M$$

Some Other Useful Properties

- A 2-D LSI system $h(n_1, n_2)$ is **stable, if**

$$\sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} |h(n_1, n_2)| < \infty$$

- A 2-D LSI system $h(n_1, n_2)$ is **causal**, if

$$h(n_1, n_2) = 0, \quad n_1 < 0, \ or \ n_2 < 0$$

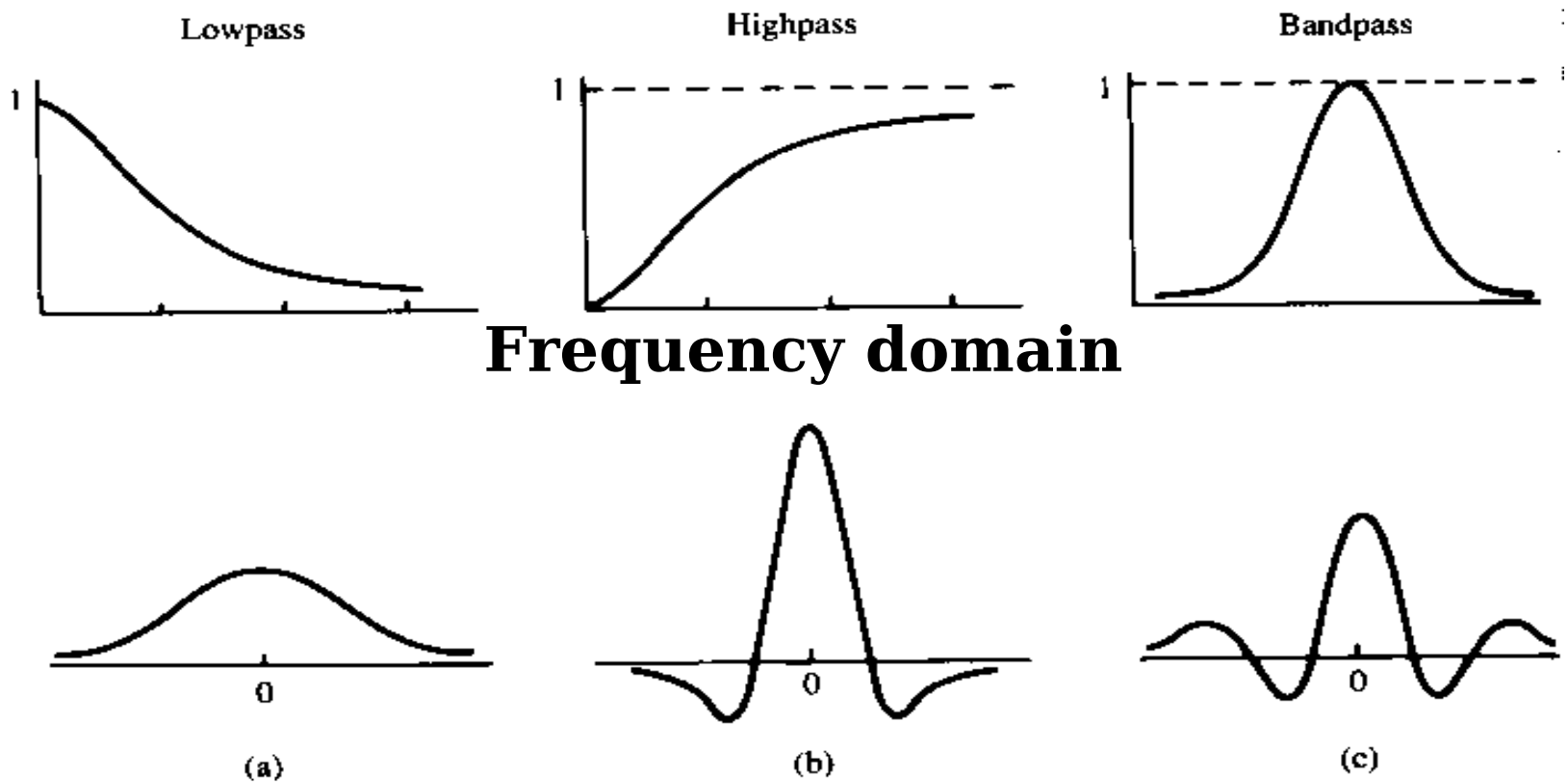
- Typical **Lowpass** Filter: done by averaging.
- Typical **Highpass** Filter: done by differencing.

2-D Convolution in Image Processing

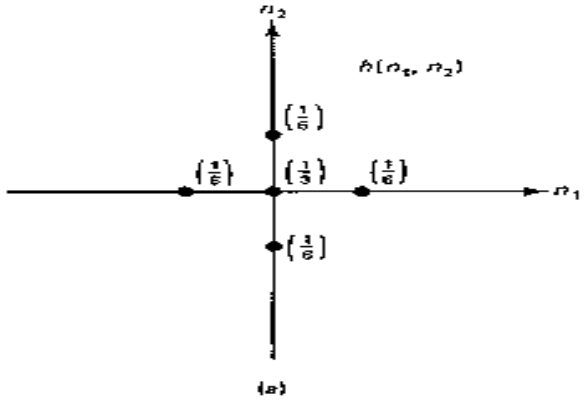
- Most filters used in image processing have coefficients symmetrical with respect to the center. In these cases, in performing the 2-D convolution, we don't need to flip $h(n_1, n_2)$.

- When the mask is partly outside the image, we often assume the values of the pixels outside the image have the same values

1-D Filters


Lowpass
Highpass
Bandpass
1
1
1
Frequency domain
0
0
0
(a)
(b)
(c)
Spatial domain

A 2-D Lowpass Filter



$$\begin{bmatrix} 0 & 1/6 & 0 \\ 1/6 & 1/3 & 1/6 \\ 0 & 1/6 & 0 \end{bmatrix}$$

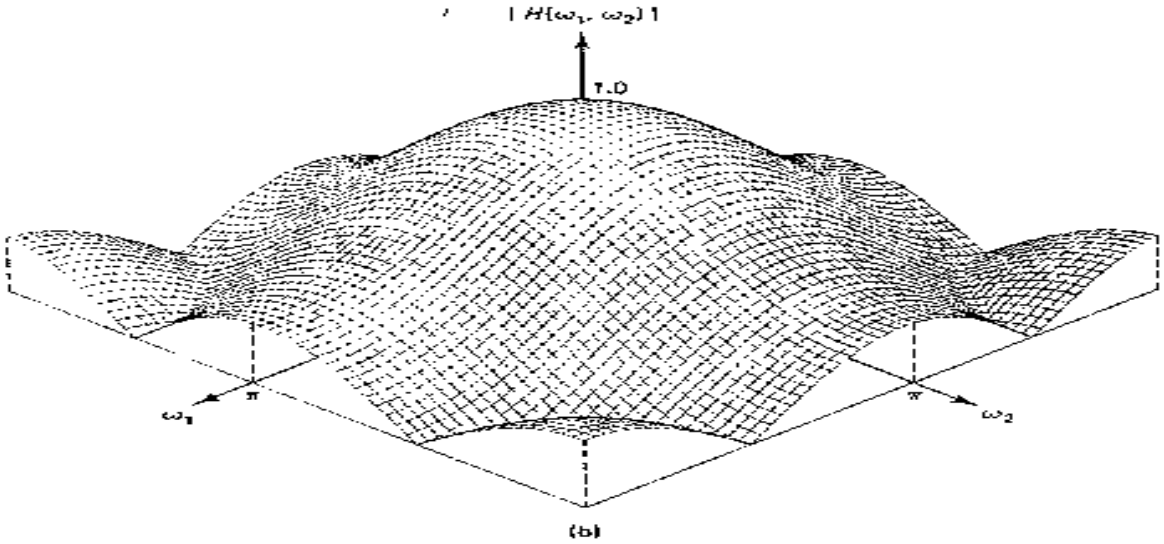


Image after Lowpass Filtering

Lowpass Filtering for Noise Smoothing

- Most image energy is in its **low frequency components** (high spatial correlation among neighboring pixels), while most image noise is **wideband** (e.g., white Gaussian noise) -- calls for lowpass filtering (LPF) (it cannot remove the noise in the low frequency range though !).
- LPFs cause **blurring**, which is critical for images with sharp edges, but not critical for images of smooth contrast.
- **Another use:** remove blocky effect created in lossy image compression.

Some Typical LPFs

- Averaging LPFs:

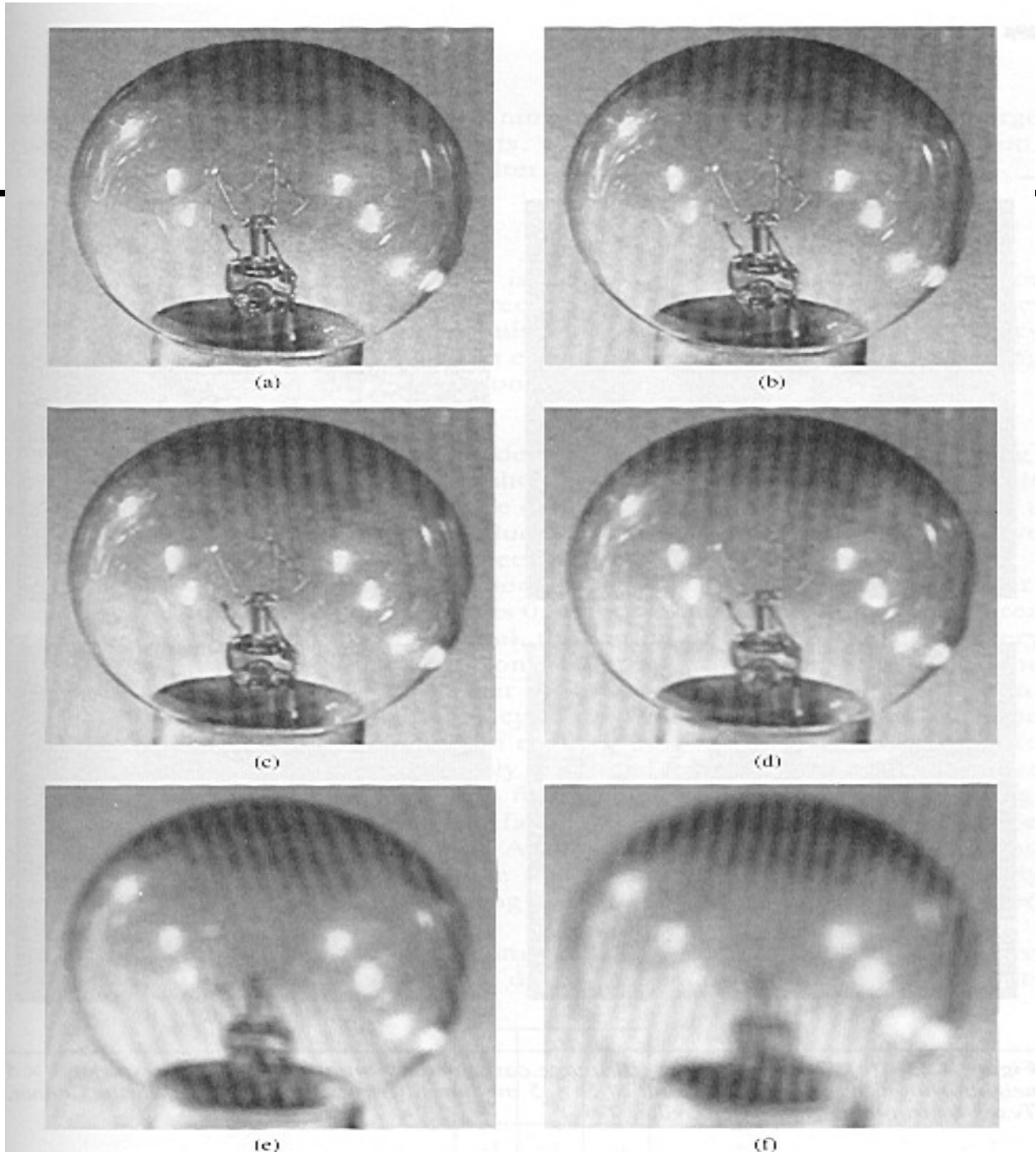
$$\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix} \qquad \frac{1}{25}\begin{bmatrix}1 & 1 & 1 & 1 & 1\\ 1 & 1 & 1 & 1 & 1\\ 1 & 1 & 1 & 1 & 1\\ 1 & 1 & 1 & 1 & 1\\ 1 & 1 & 1 & 1 & 1\end{bmatrix}$$

- Some other (Gaussian) LPFs:

$$\frac{1}{10}\begin{bmatrix}1 & 1 & 1\\ 1 & 2 & 1\\ 1 & 1 & 1\end{bmatrix} \qquad \frac{1}{16}\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix}$$

Lowpass Filtered Images

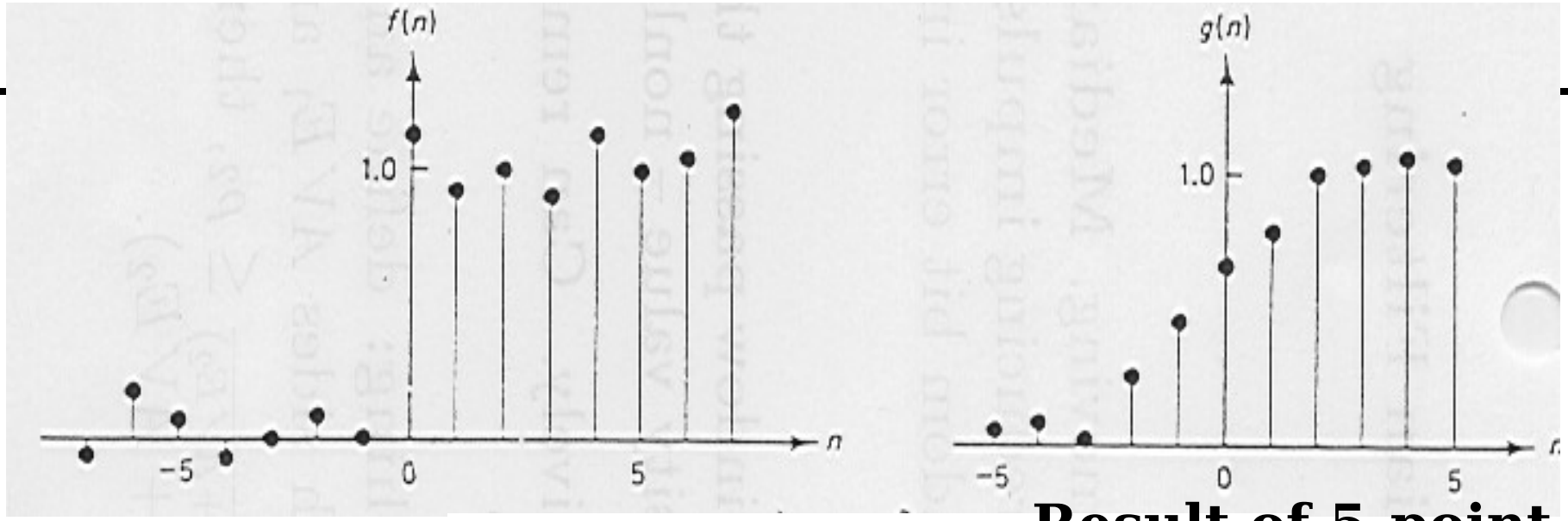
- Average filter of size: 3x3, 5x5, 7x7, 15x15, and 25x25.



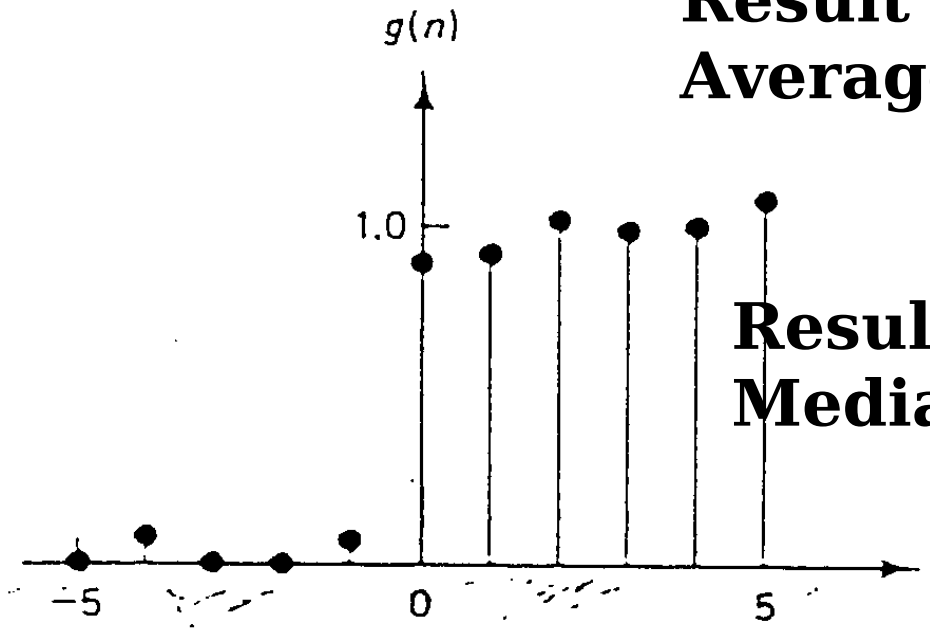
Median Filtering

- To avoid blurring in noise removing. **Median filtering** is a nonlinear process especially useful for reducing **impulse**, or **salt-and-pepper noise**.
- It preserves **edges**.
- **Median filtering**: with a sliding window passing through the image, choose the **median** (middle) intensity value -- nonlinear.
- Sorting can be done recursively.

Median Filters Preserve Edges

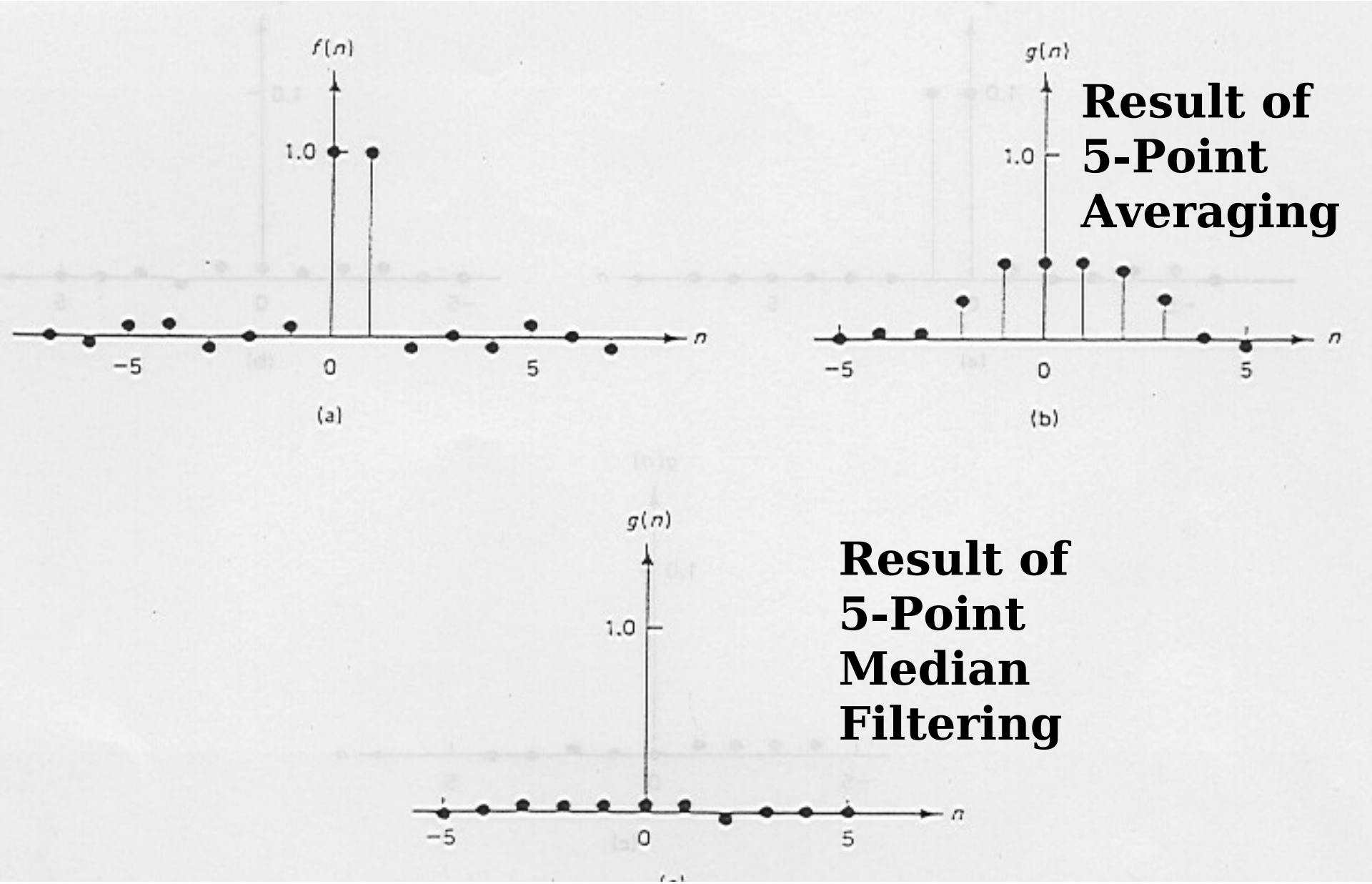


Result of 5-point Average Filter

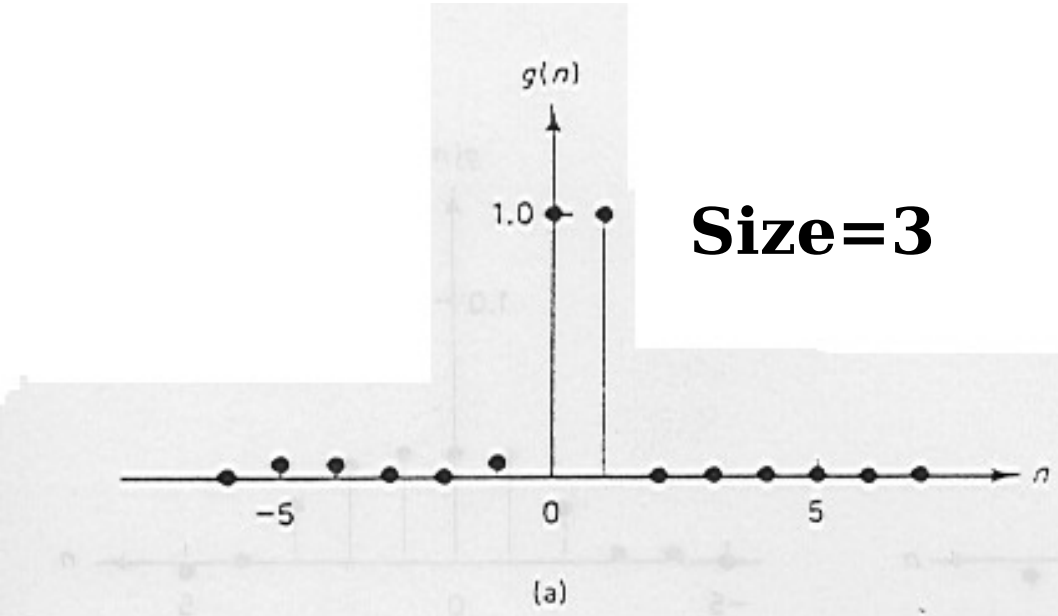


Result of 5-Point Median Filter

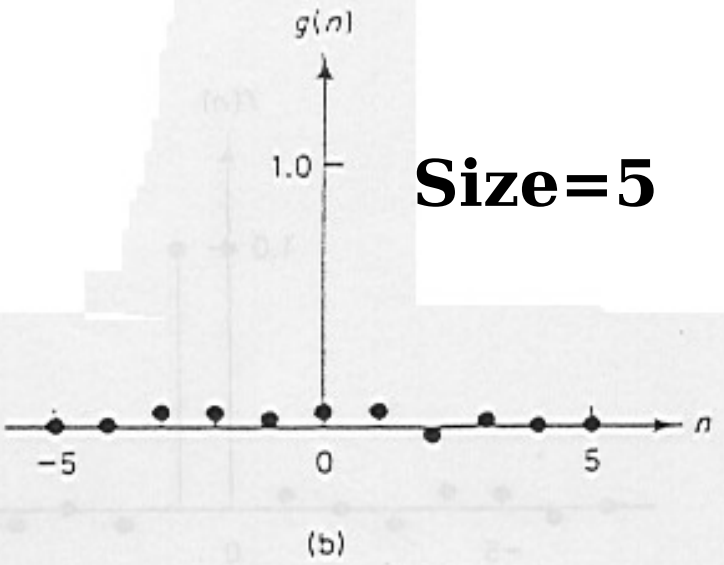
Noise


f(n)
1.0
−5
0
5
n
(a)
g(n)
1.0
−5
0
5
n
(b)
Result of 5-Point Averaging
g(n)
1.0
−5
0
5
n
Result of 5-Point Median Filtering

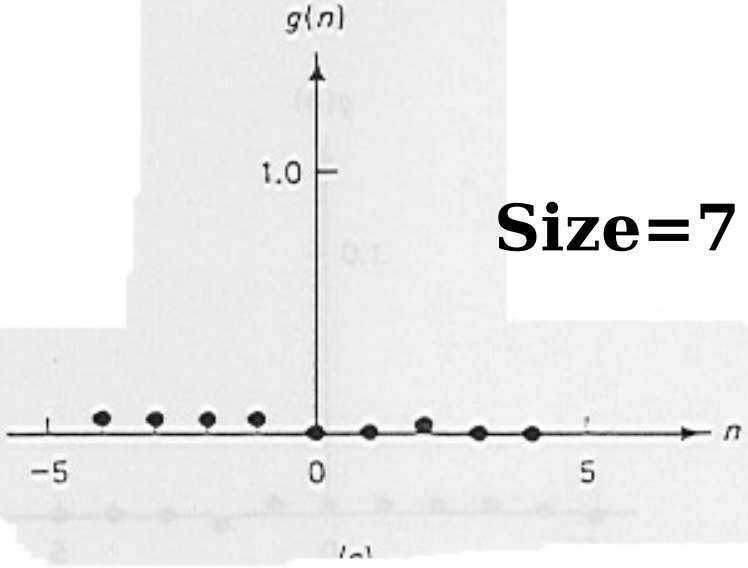
Effect of Median Filter Window Sizes



Size=3

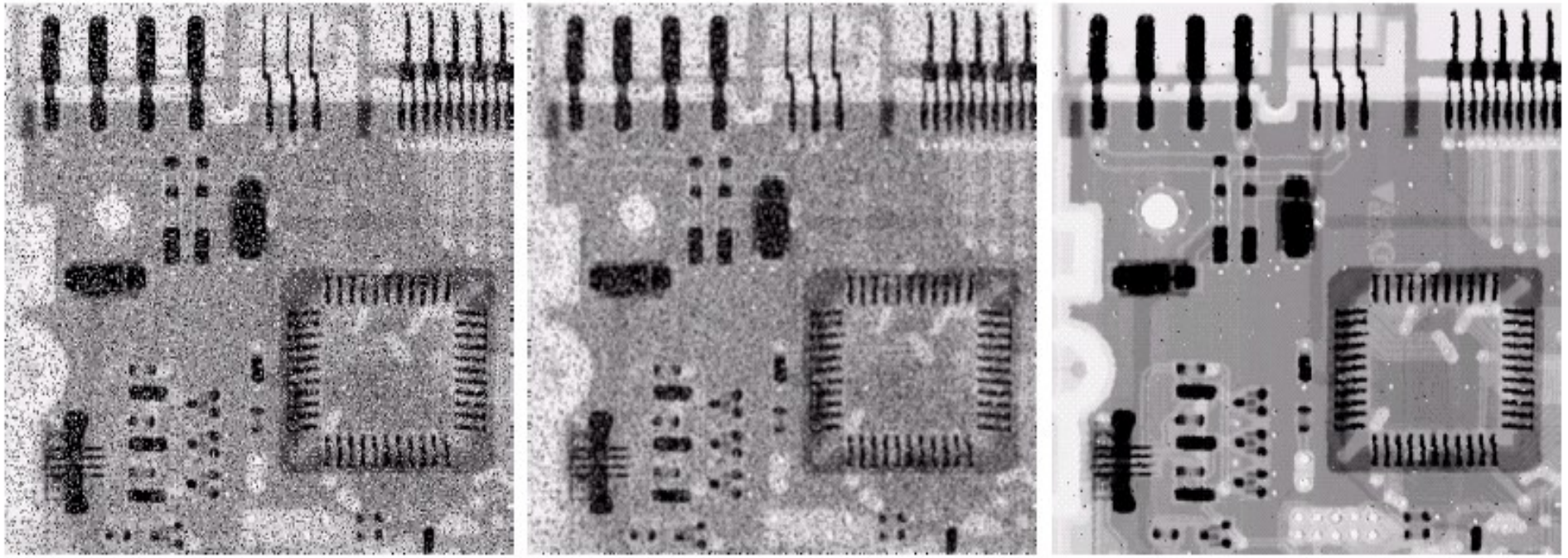


Size=5



Size=7

Averaging Mask vs. Median Filter



a b c

FIGURE 3.37 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3×3 averaging mask. (c) Noise reduction with a 3×3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

Impulse Noise Removing

(a)

(b)

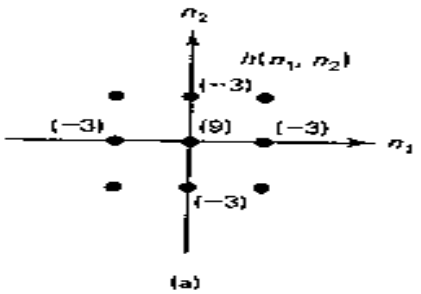
(c)

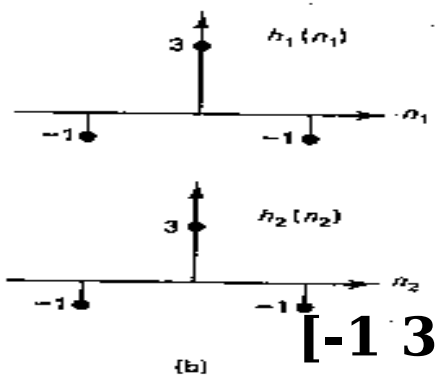
(d)

a) original
b) impulse noise
c) 5x5 averaging
d) 5x5 median

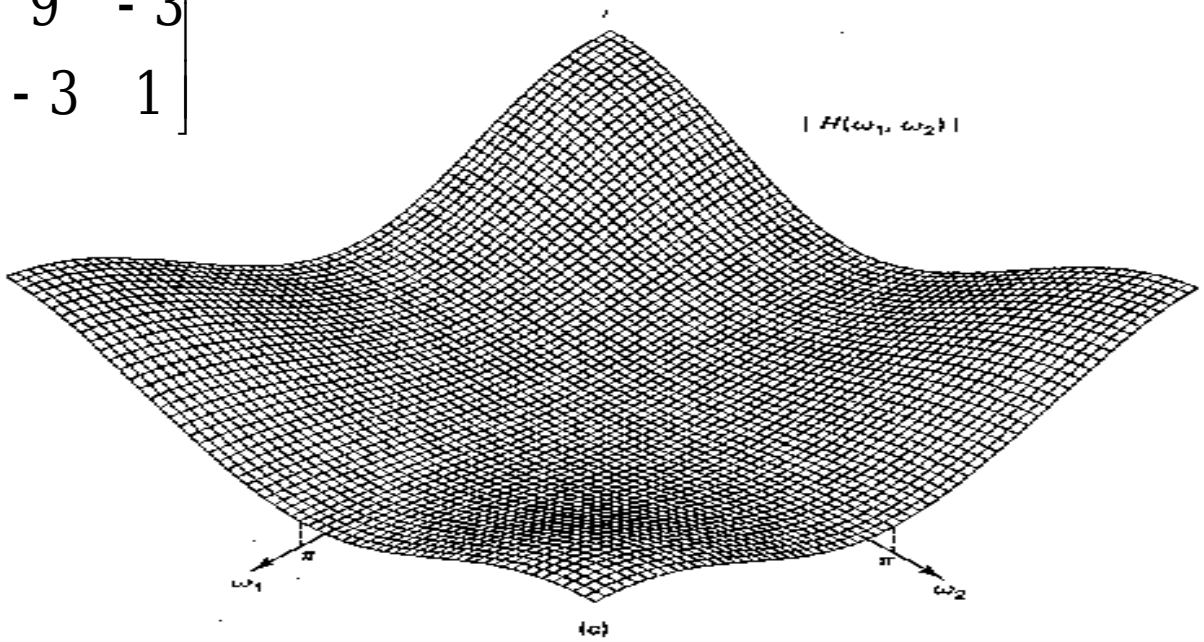
A Highpass Filter

$$\begin{bmatrix} -1 \\ 3 \\ -1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -1 \end{bmatrix} = \begin{bmatrix} 1 & -3 & 1 \\ -3 & 9 & -3 \\ 1 & -3 & 1 \end{bmatrix}$$





[-1 3 -1]

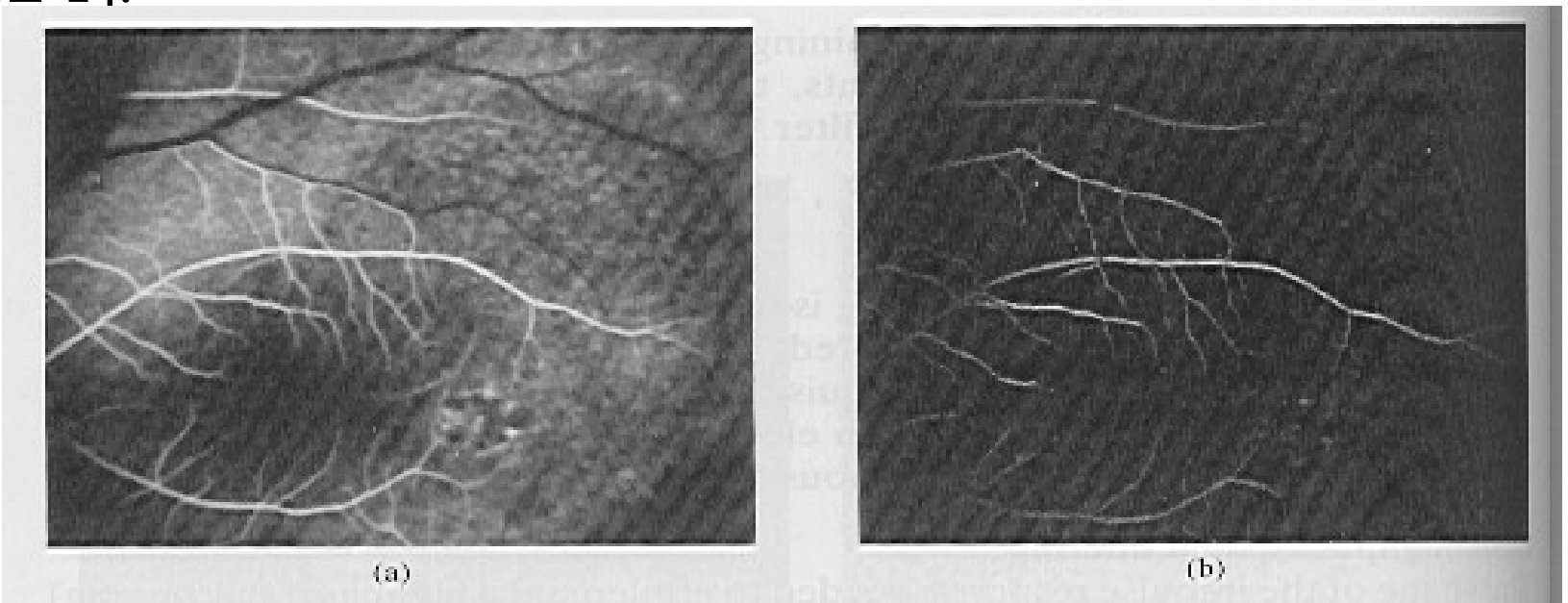


Highpass Filters

- A **basic highpass spatial filtering**

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

- Note that the sum of coefficients is **0**! The DC term of H(w1,w2) is zero, i.e., the global contrast of the images is reduced.
- Results need level-shift and scaling or clipping to remain in [0,L-1].



Derivative Operators

- An **averaging** is analogous to **"integration"**, **"differentiation"** can be expected to have the opposite effect: **sharpening**.
- **Differentiation** (gradient) of $f(x,y)$:

$$\nabla f = \begin{bmatrix} \frac{\partial f(x,y)}{\partial x} \\ \frac{\partial f(x,y)}{\partial y} \end{bmatrix} \qquad mag(\nabla f) = [(\frac{\partial f}{\partial x})^2 + (\frac{\partial f}{\partial y})^2]^{1/2}$$

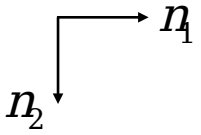
$$or \quad mag(\nabla f) \approx |\frac{\partial f}{\partial x}| + |\frac{\partial f}{\partial y}|$$

- In **discrete** domain:

$$\frac{\partial f(n_1,n_2)}{\partial n_1} \approx f(n_1,n_2) - f(n_1-1,n_2)$$

$$\approx f(n_1+1,n_2) - f(n_1,n_2)$$

$$\approx \frac{1}{2}[f(n_1+1,n_2) - f(n_1-1,n_2)]$$



Derivative Operators

- Typically, we want to average over several neighboring rows, i.e.,

$$\frac{\partial f(n_1, n_2)}{\partial n_1} \approx \left[f(n_1 + 1, n_2 - 1) - f(n_1 - 1, n_2 - 1) \right]$$
$$+ \left[f(n_1 + 1, n_2) - f(n_1 - 1, n_2) \right]$$
$$+ \left[f(n_1 + 1, n_2 + 1) - f(n_1 - 1, n_2 + 1) \right]$$

- These result in **Prewitt** operators:

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

Prewitt Operator

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$ **operate on**

$$\begin{bmatrix} f(n_1-1, n_2-1) & f(n_1, n_2-1) & f(n_1+1, n_2-1) \\ f(n_1-1, n_2) & f(n_1, n_2) & f(n_1+1, n_2) \\ f(n_1-1, n_2+1) & f(n_1, n_2+1) & f(n_1+1, n_2+1) \end{bmatrix}$$

$$\begin{aligned} = & \left[f(n_1+1, n_2-1) - f(n_1-1, n_2-1) \right] \\ + & \left[f(n_1+1, n_2) - f(n_1-1, n_2) \right] \\ + & \left[f(n_1+1, n_2+1) - f(n_1-1, n_2+1) \right] \end{aligned}$$

Derivative Operators

- How about emphasize the center row more:

$$\frac{\partial f(n_1, n_2)}{\partial n_1} \approx \left[f(n_1 + 1, n_2 - 1) - f(n_1 - 1, n_2 - 1) \right]$$
$$+ 2\left[f(n_1 + 1, n_2) - f(n_1 - 1, n_2) \right]$$
$$+ \left[f(n_1 + 1, n_2 + 1) - f(n_1 - 1, n_2 + 1) \right]$$

- These result in **Sobel** operators:

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Diagonal Derivative Operators

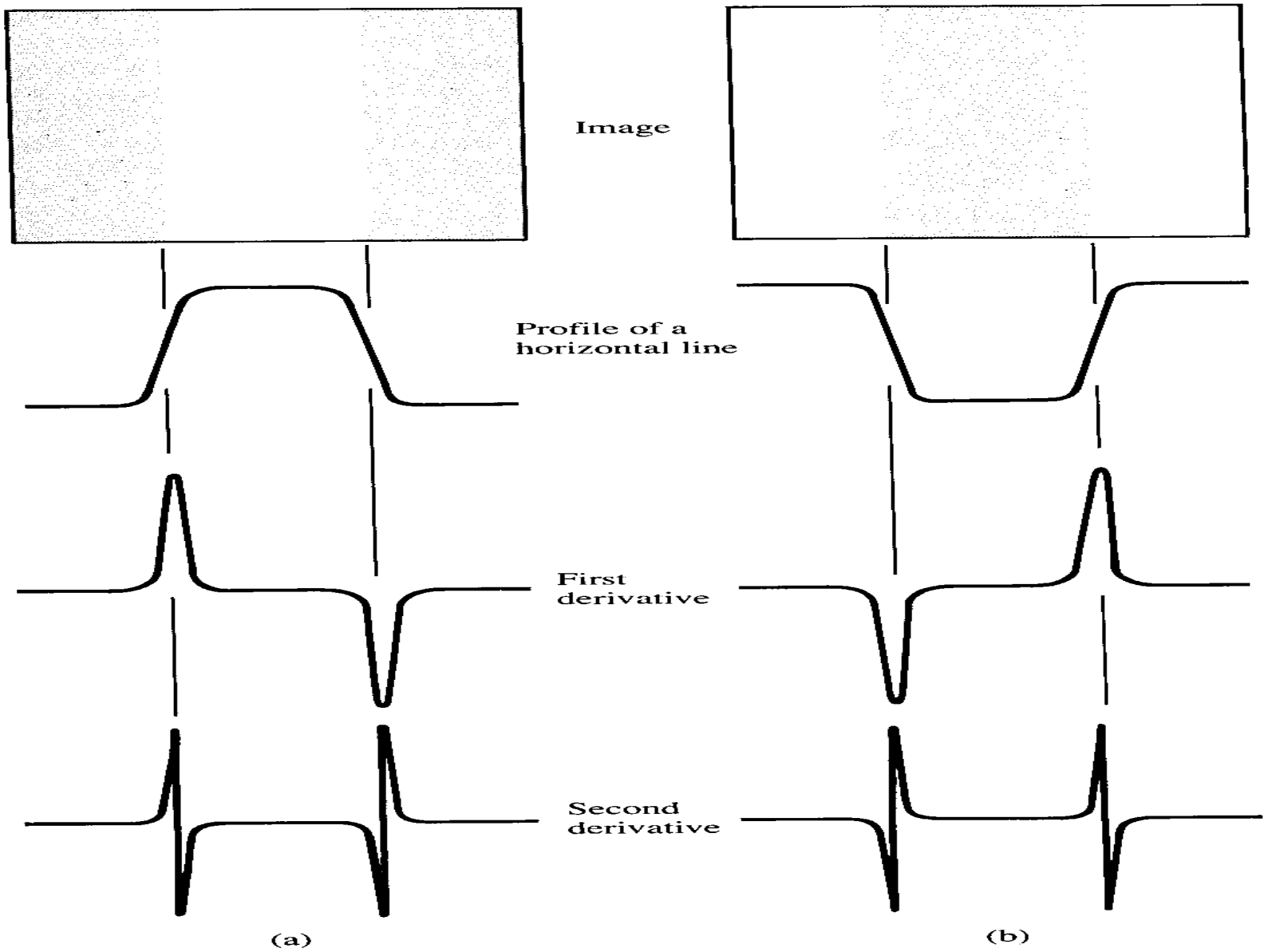
- We can also deal with both derivatives simultaneously:

$$\frac{\partial f(n_1,n_2)}{\partial n_1}+\frac{\partial f(n_1,n_2)}{\partial n_2}\approx\left[\, f(n_1,n_2)-\; f(n_1+1,n_2+1)\right]$$

- This results in **Robert** operators:

$$\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}\qquad\begin{bmatrix}0 & 1\\ -1 & 0\end{bmatrix}$$

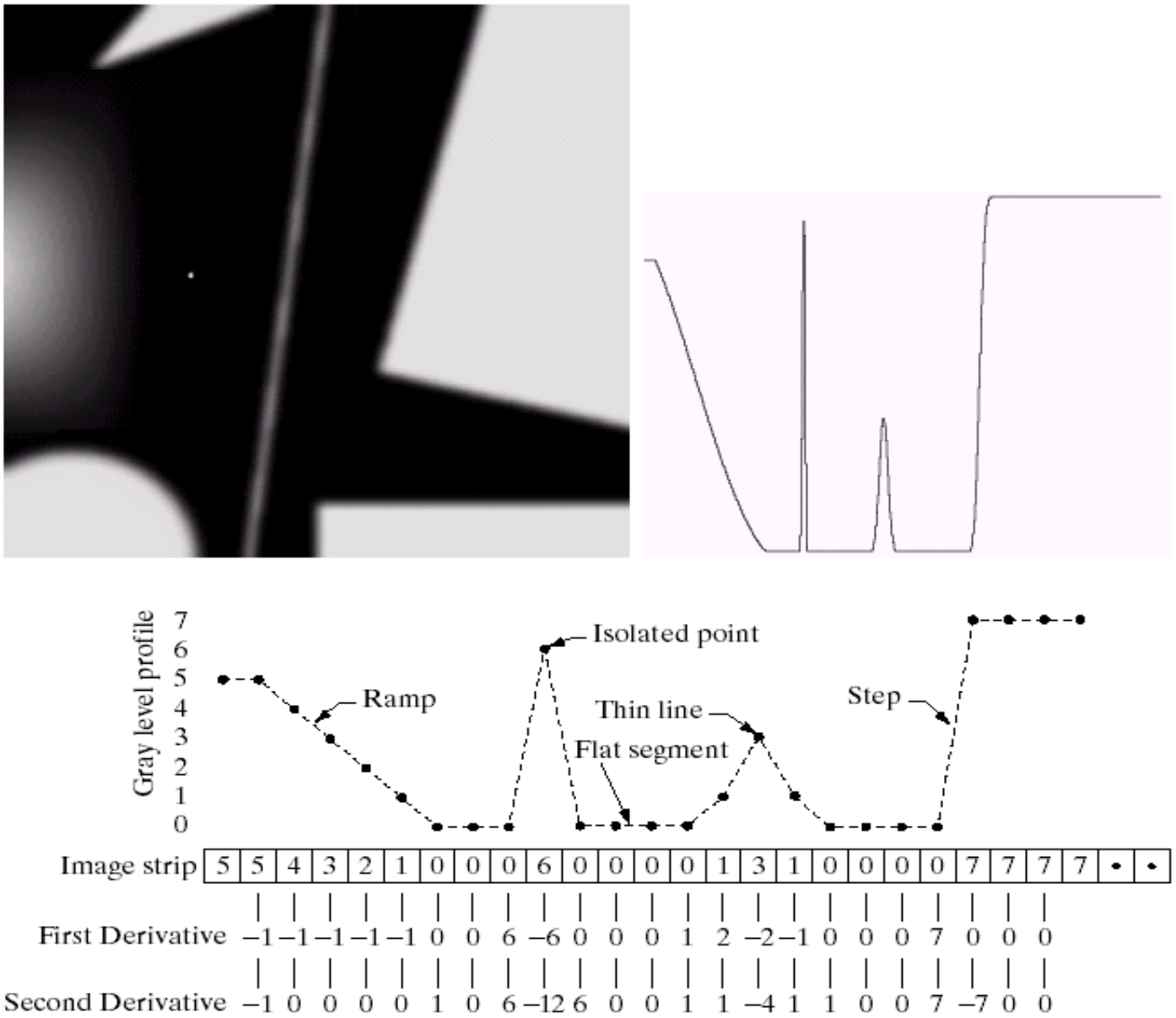
Effects of 1st & 2nd Derivatives



First and Second

a b
c

FIGURE 3.38
(a) A simple image. (b) 1-D horizontal gray-level profile along the center of the image and including the isolated noise point.
(c) Simplified profile (the points are joined by dashed lines to simplify interpretation).



Laplacian Operators: 2nd Derivative

$$\nabla^2 f(n_1, n_2) \cong \frac{\partial^2 f(n_1, n_2)}{\partial n_1^2} + \frac{\partial^2 f(n_1, n_2)}{\partial n_2^2}$$

- Difference Eq. approximation of Laplacian:

$$\frac{\partial f(n_1, n_2)}{\partial n_1} \approx f(n_1 + 1, n_2) - f(n_1, n_2)$$

$$\frac{\partial^2 f(n_1, n_2)}{\partial n_1^2} \approx \left[f(n_1 + 1, n_2) - f(n_1, n_2) \right] - \left[f(n_1, n_2) - f(n_1 - 1, n_2) \right]$$

$$= f(n_1 + 1, n_2) - 2 f(n_1, n_2) + f(n_1 - 1, n_2)$$

$$\nabla^2 f(n_1, n_2) \cong f(n_1 + 1, n_2) + f(n_1 - 1, n_2) +$$

$$f(n_1, n_2 + 1) + f(n_1, n_2 - 1) - 4 f(n_1, n_2)$$

Laplacian Operators

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 2 & -1 \\ 2 & -4 & 2 \\ -1 & 2 & -1 \end{bmatrix}$$

- The edges are indicated by **zero crossings** (which offer more reliable edge location), instead of **large gradients**.

Enhancement Using Laplacian Operators

$$g(x,y) = \begin{cases} f(x,y) - \nabla^2 f(x,y), & \text{if center Laplacian coeff is negativ} \\ f(x,y) + \nabla^2 f(x,y), & \text{if center Laplacian coeff is positiv} \end{cases}$$

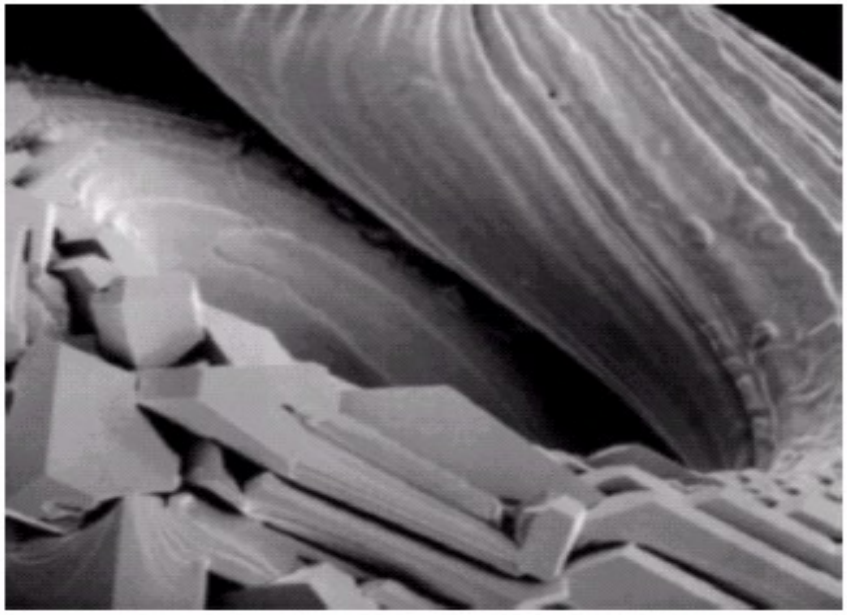
$$g(x,y) \cong f(x,y) - [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)] = 5f(x,y) - [f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1)]$$

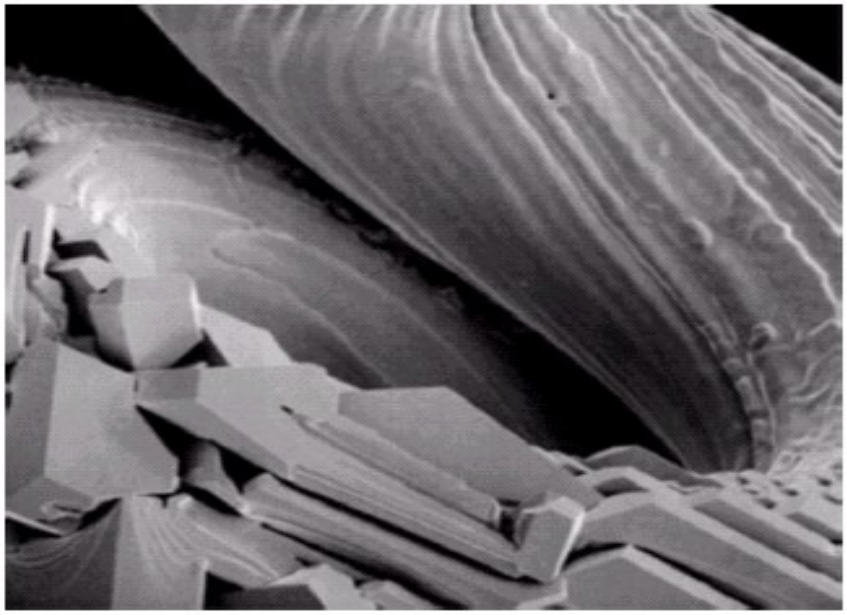
a b
c d

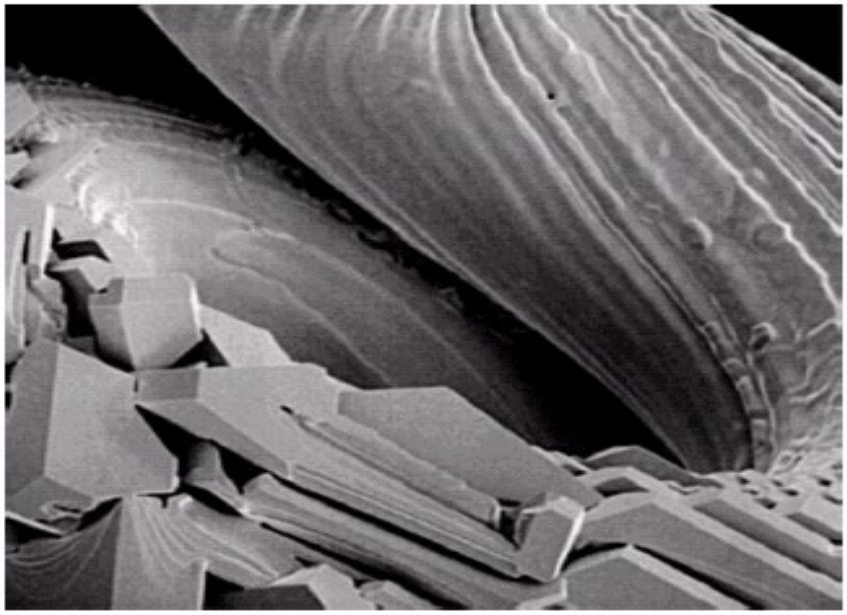
FIGURE 3.40
(a) Image of the North Pole of the moon.
(b) Laplacian-filtered image.
(c) Laplacian image scaled for display purposes.
(d) Image enhanced by using Eq. (3.7-5).
(Original image courtesy of NASA.)

0	−1	0
−1	5	−1
0	−1	0

−1	−1	−1
−1	9	−1
−1	−1	−1







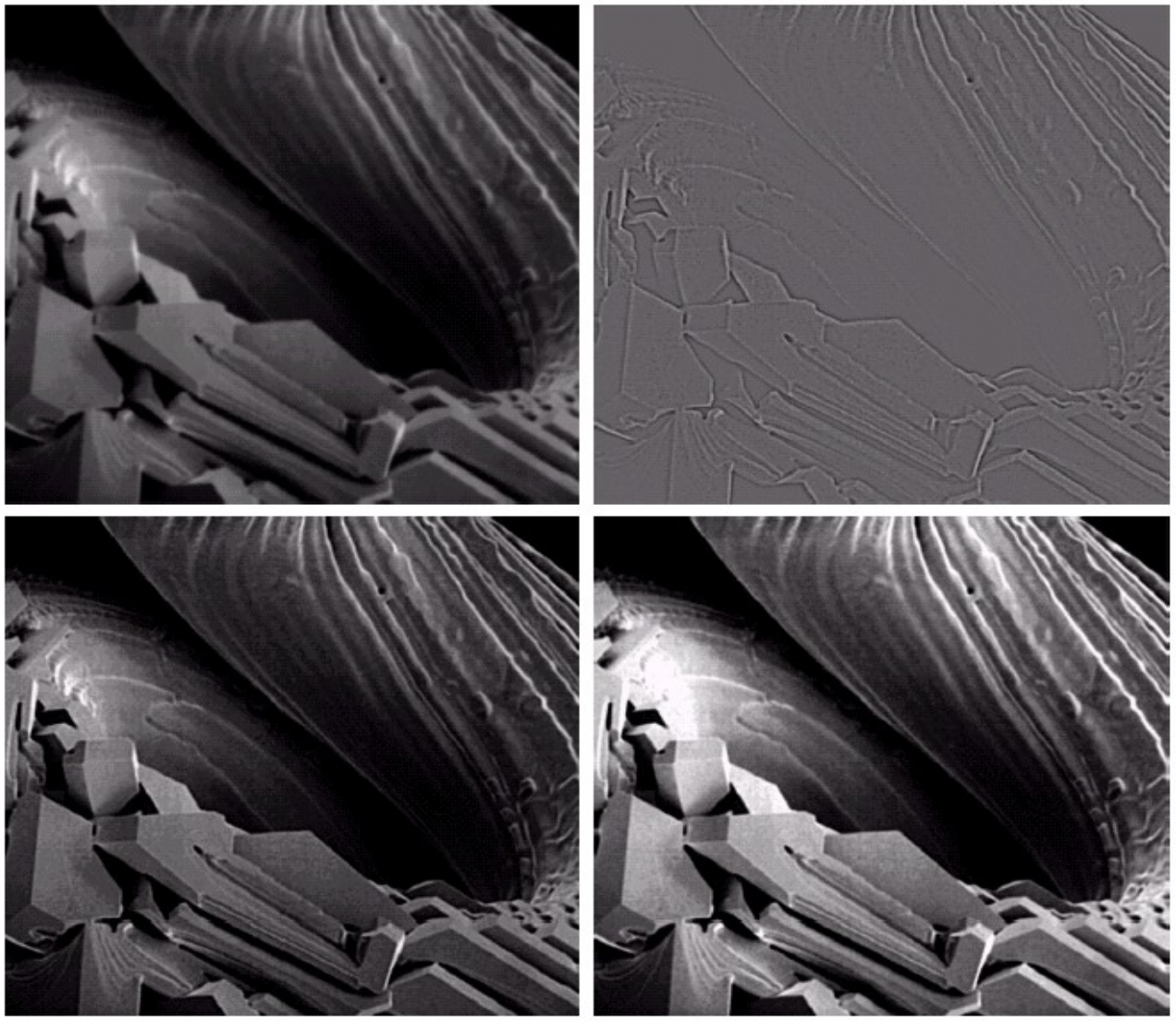
a b c
d e

FIGURE 3.41 (a) Composite Laplacian mask. (b) A second composite mask. (c) Scanning electron microscope image. (d) and (e) Results of filtering with the masks in (a) and (b), respectively. Note how much sharper (e) is than (d). (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

High-Boost Filtering

- With $A \geq 1$,

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & A+8 & -1 \\ -1 & -1 & -1 \end{bmatrix} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & A & 0 \\ 0 & 0 & 0 \end{bmatrix}$$



a b
c d

FIGURE 3.43
(a) Same as Fig. 3.41(c), but darker.
(a) Laplacian of (a) computed with the mask in Fig. 3.42(b) using $A = 0$.
(c) Laplacian enhanced image using the mask in Fig. 3.42(b) with $A = 1$. (d) Same as (c), but using $A = 1.7$.

High-Boost Filtering

- With $A \geq 1$,

$$\frac{1}{9}\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9A-1 & -1 \\ -1 & -1 & -1 \end{bmatrix} = \frac{1}{9}\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & A-1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

A-1)(original)
- Highpass

(a) original
(b) A=1.1
(c) A=1.15
(d) A=1.2

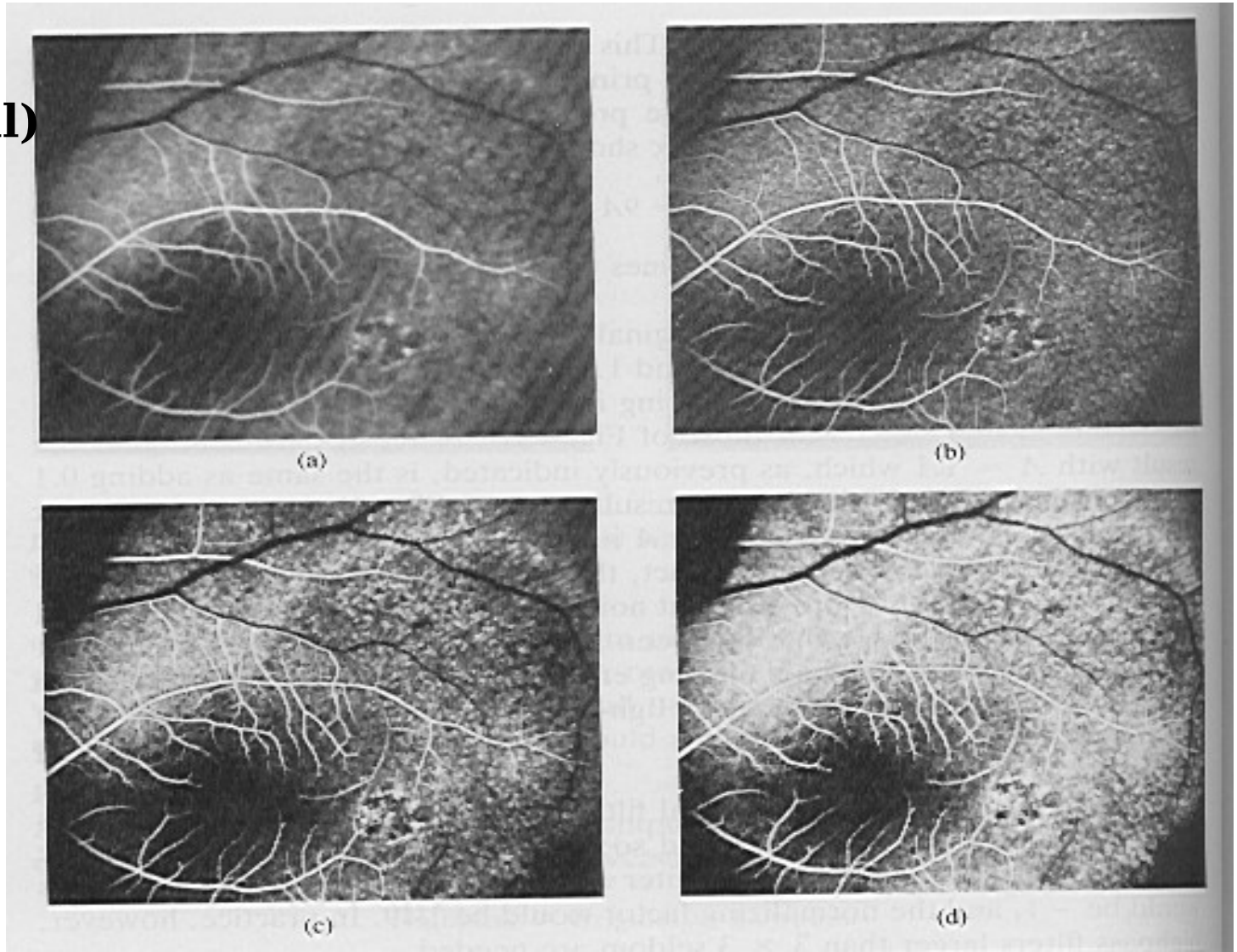


Image after High-Boost Filtering